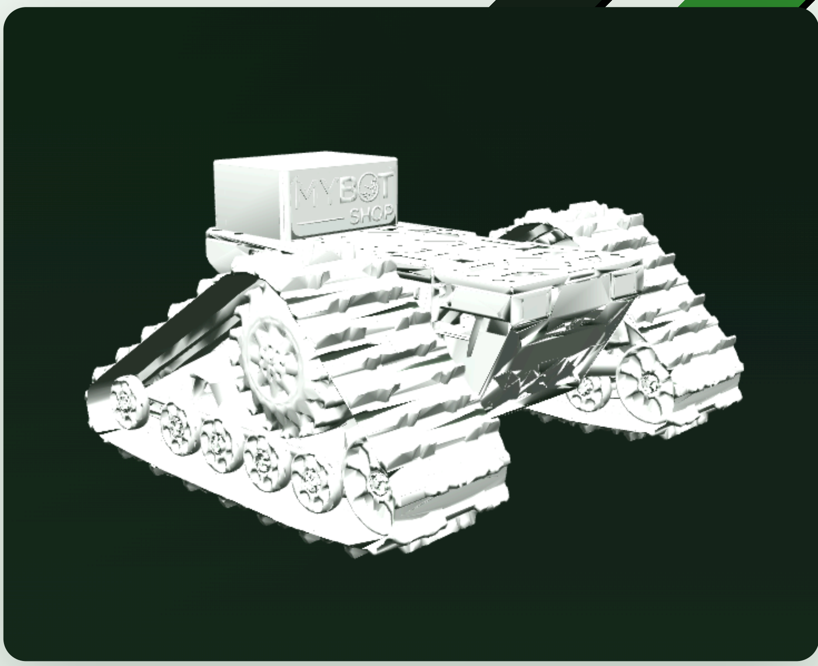
MYBOTSHOP

ROBOTICS

ROVO3 Tracked Robot

User Manual



CONTACT

info@mybotshop.de

support@mybotshop.de

mybotshop.de

Contents

1 Attention

Attention: Read this document carefully before operating the robot. In case of doubt, consult the manufacturer (HAWE Mattro, for the underlying ROVO drive platform) or the integrator/distributor (MYBOTSHOP GmbH) of the assemblies used. The robot must not be put into operation before clarification. Contact information is available in Chapter 9, Getting Help & RMA.

Do not allow persons who are not familiar with the robot or these instructions to operate it. Robots are dangerous in the hands of untrained users.

Do not operate the robot if you are intoxicated, under the influence of drugs, or unable to concentrate. Familiarize yourself with the operating-mode switch, gear system and emergency-stop procedure in this manual before switching the robot on.

Caution: ROVO3 is a ~350 kg tracked ground vehicle with a top speed of 13.4 km/h, intended for outdoor and rough-terrain autonomous mobile robot (AMR) use. Keep a clear area around the robot while it is powered and under active control. Units pre-configured with ROS 2 start in *ROS2 Mode* by default, in which the physical ROVO controller's joystick and gears are disabled and only its emergency stop remains active — motion is only possible through the ROS 2 driver. This manual covers that ROS 2 / R&D-kit layer as an extension of the base manufacturer documentation shipped with the robot.

2 Disclaimer

The ROVO3 documented here is built on the HAWE Mattro ROVO tracked-drive platform, integrated and supported by *MYBOTSHOP GmbH*. Basic ROS 2 understanding is required to operate the software stack covered in this manual — if you are not familiar with ROS 2, we recommend checking the ROS 2 documentation first.

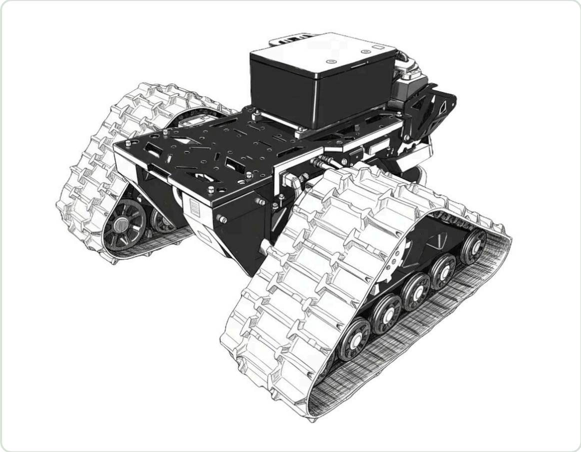
The client acknowledges and agrees that any information or materials provided by MYBOTSHOP GmbH are for R&D and development purposes only. Any services are provided “AS IS” and without any representation or warranty of any kind, express or implied, including but not limited to any warranty of merchantability, fitness for a particular purpose, non-infringement, or any other warranty.

This manual is an extension of the robot manufacturer documentation for ROS 2 users, and covers the R&D kit. For base power-on and hardware operation of the ROVO3 outside of ROS 2, refer to the manufacturer product documentation shipped with the robot (customer-only; see Chapter 3). Every joint in the robot's URDF description is fixed — locomotion is driven entirely by the skid-steer track hardware and, in simulation, by Gazebo's TrackedVehicle plugin, not by articulated joints. Always confirm the operating mode (ROS2 Mode vs. Radio Controller Mode) before attempting to drive the robot by any method. For terms, policies and compliance with local laws and regulations, refer to the MYBOTSHOP website and the RMA/support process in Chapter 9.

MYBOTSHOP GmbH shall not be liable for any damages, including but not limited to direct, indirect, special, incidental, or consequential damages, arising out of or in connection with the use or inability to use the information or materials in this document. This limitation on liability shall apply regardless of the form of action, whether in contract, tort, or otherwise.

3 About the ROVO3

ROVO3 Tracked Robot



MYBOTSHOP ROVO3 Tracked Robot Platform

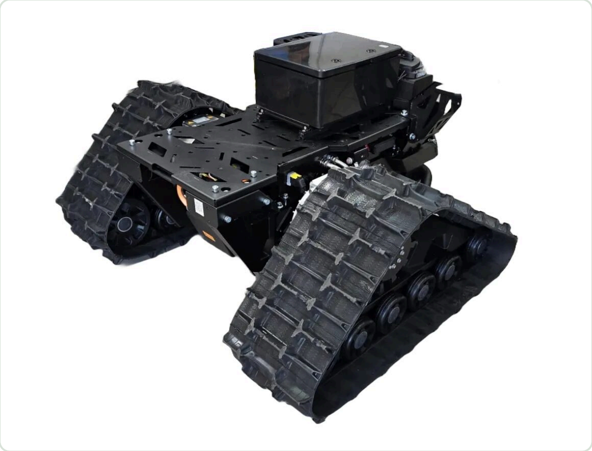
The MYBOTSHOP ROVO3 is a rugged tracked robot platform designed for autonomous mobile robot (AMR) applications. Built for outdoor and challenging terrain environments, the ROVO3 combines robust tracked locomotion with advanced ROS2 integration for research, industrial, and defense applications.

Key Features:

- Tracked chassis for superior terrain traversal
- ROS2 Jazzy support with Gazebo Harmonic simulation
- Integrated IMU and odometry
- Web-based control interface
- Support for LiDAR, depth cameras, and GPS
- CAN bus communication for platform control

Current Software Version: v2.0 (December 2025)

Introduction



MYBOTSHOP ROVO3 Tracked Robot Platform

The MYBOTSHOP ROVO3 is a rugged tracked robot platform designed for autonomous mobile robot (AMR) applications. The tracked chassis provides superior terrain traversal capabilities compared to wheeled platforms, making it ideal for outdoor environments, rough terrain, and challenging surfaces.

Quick Links

Resource	Link
MYBOTSHOP Website	mybotshop.de
ROVO3 ROS Documentation	docs.mybotshop.de
Robot Webserver	192.168.131.1:9000
ROS2 Jazzy Documentation	docs.ros.org/jazzy
Gazebo Harmonic Docs	gazebosim.org
Nav2 Documentation	nav2.org

External Dependencies

Package	Documentation
Ouster LiDAR Driver	ouster-ros
Intel RealSense ROS2	realsense-ros
Teleop Twist Keyboard	teleop_twist_keyboard
Nav2 Navigation	navigation2
ros_gz Bridge	ros_gz

Current Available Features

Feature	Status	Notes
ROVO3 Visualization (RViz2)	Available	
ROVO3 Base High Level Driver	Available	Joint State, IMU, CMD, Odometry
ROVO3 Description (URDF/Xacro)	Available	
ROVO3 Lidar (Ouster)	Available	
ROVO3 Depth Camera (Intel Realsense D435)	Available	
ROVO3 Autostart Services	Available	
ROVO3 Twist Multiplexer	Available	
ROVO3 SLAM	Available	
ROVO3 Odometry Navigation	Available	
ROVO3 Map Navigation	Available	
ROVO3 Webserver	Available	
ROVO3 Gazebo Harmonic Simulation	Available	

Manuals

Note: Product documentation and manuals are available for customers only. Please contact MYBOTSHOP for access to the following documents: ROVO3 Flyer (DE/EN) ROVO3 Assembly Manual ROVO3 Product Documentation ROVO3 CAN Control Guide ROVO3 CAN Interface Documentation

ROVO Platform

The ROVO3 is based on the HAWE Mattro ROVO platform, a fully electric tracked drive system designed for off-highway autonomous applications.

Specifications

Mechanical

Parameter	Value
Chassis Type	Tracked
Drive System	Dual Track Drive
IP Rating	Weather Resistant

Electrical

Parameter	Value
Battery Type	Lithium
Communication Interface	CAN Bus
CAN Bitrate	500 kbit/s

Base Software

Parameter	Value
Operating System	Ubuntu 24.04 LTS
ROS Distribution	ROS2 Jazzy
Simulator	Gazebo Harmonic
Navigation Stack	Nav2

Transport

Unpacking

When unpacking the ROVO3:

1. Carefully remove all packaging materials
2. Inspect the robot for any shipping damage
3. Verify all components against the packing list
4. Store packaging materials for future transport

Packing

For transport or storage:

1. Power off the robot and disconnect all cables
2. Secure any loose components
3. Use original packaging if available
4. Ensure adequate padding for protection

4 Quick Start – Modes, Power On/Off & Network

Pre-requisites

Before operating the ROVO3, ensure you have:

- Read and understood the safety guidelines
- Access to the robot's network (wired or wireless)
- A computer with ROS2 Jazzy installed (for visualization)

Important: This manual is for the R&D kit and contains additional information about the ROVO3 tracked robot. For powering on and basic usage of the ROVO3, refer to the ROVO3 product documentation provided with the robot. This is an extension of the robot manufacturer documentation for ROS2 users.

Operating Modes

Important: The ROVO3 has 2 Modes: Radio Controller Mode - For this mode please reference the ROVO3 manufacturer's manual. ROS2 Mode - This is the default for robots that are pre-configured with ROS. In this mode the joystick of the ROVO3 controller nor the gears work. Only the emergency stop button works. The robot has to be controlled via ROS (i.e., Logitech controller, webserver, or ROS2 services). To re-enable radio control mode, turn off the rovo-platform via ROS2 services and this will re-enable the radio controller.

Robot App

The ROVO3 comes with pre-installed ROS2 software. The robot's onboard computer runs Ubuntu 24.04 LTS with ROS2 Jazzy. Core drivers and services start automatically on boot via systemd services.

Robot Webserver

The ROVO3 features a web-based control interface accessible at:

URL: http://192.168.131.1:9000

Default Credentials:

- Username: admin
- Password: mybotshop

The webserver provides:

- Real-time robot visualization (3D model)
- Joystick teleoperation

- Service management (start/stop ROS nodes)
- Map visualization
- Battery monitoring
- Rosbag recording
- VNC remote desktop access

Charging Robot

To charge the ROVO3:

1. Power off the robot
2. Connect the charger to the charging port
3. Monitor the charging status via the charger LED
4. Disconnect when fully charged

Charging Controller

The ROVO3 uses an integrated Battery Management System (BMS) for safe charging and discharging operations.

Emergency Stop

The ROVO3 is equipped with emergency stop buttons. When pressed:

- All motor power is immediately cut
- The robot enters a safe parking state
- Services continue running but motion commands are ignored

To resume operation after emergency stop:

1. Clear the emergency condition
2. Release the emergency stop button
3. Reset the gear to neutral, then re-engage

Power On

To power on the ROVO3:

1. Release the emergency stop button by turning it clockwise
2. Press the power button to turn on the robot
3. Wait for the robot to boot up completely (approximately 1 minute)
4. Release the emergency stop button by turning it clockwise on the remote controller
5. Power on the ROVO remote controller
6. Press the vibration button on the Logitech controller to ensure it is powered on

7. Press up arrow on the Logitech controller to go to a higher gear (you should see the gear change in the ROVO Controller)
8. Now with LB + Left Joystick you can move the robot forward/backward and LB + Right Joystick for yaw left/right
9. Press down arrow on the Logitech controller to go to a lower gear

Warning: Please note that the robot ROS2 driver cannot operate without the powering of the ROVO3 Remote Controller for safety reasons and inclusion of the emergency stop functionality. The robot by default starts in ROS2 mode. To deactivate it simply go to the webserver and disable the rovo services. While in ROS2 mode only the emergency stop will work from the ROVO Controller.

Power Off

To safely power off the ROVO3:

1. Set the gear to neutral (gear 0)
2. Stop all running applications
3. Press the main power switch to turn off

Basic Operation

Remote Control

The ROVO3 supports multiple teleoperation methods:

1. Logitech Joystick - Highest priority, requires deadman switch
2. Steamdeck Controller - Wireless control via Steamdeck
3. Web Interface - Browser-based joystick control
4. Keyboard Teleop - ROS2 teleop_twist_keyboard

Network Interface

Network Table

IP Address	Device	Username	Password
192.168.131.1	Robot MCU	robot	mybotshop
192.168.131.1:9000	Webserver	admin	mybotshop
192.168.131.150	Steamdeck	deck	mybotshop
192.168.131.200	Router	SSID	mybotshop
192.168.131.200	Router-Web	admin	Admin1232025

192.168.131.20	Ouster Lidar	x	x

Warning: Do not set your computer's IP to any of the above reserved addresses.

Note: The robot login credentials are: Username: robot Password: mybotshop

Ethernet Connection

1. Power on the robot
2. Connect a LAN cable to the robot's ethernet port (topmost port on demo units)
3. Configure your computer's network settings

Static Network Connection

For first-time connection, configure a static IP on your computer:

1. Go to Settings > Network > + (add new connection)
2. Select Manual in IPv4 settings
3. Set the following: Address: 192.168.131.51 Netmask: 24
4. Save and restart your network

Verify the connection:

```
# Check your local IP
ifconfig

# Ping the robot
ping 192.168.131.1

# SSH into the robot
ssh -X robot@192.168.131.1
# Password: mybotshop
```

5 ROS 2 Operation – Autostart, Webserver, Teleoperation & Visualization

Pre-requisites

- Safety Guidelines Robotic Manipulator Safety .
- Power On Power On .
- Network Interface Network Interface .
- Installation Robot Installation .

Autostart

Humble

Coming Soon

Jazzy

Systemd Services

The ROVO3 uses systemd services for automatic startup of ROS2 nodes.

Install Startup Services:

```
ros2 run rovo_autostart startup_installer.py
```

Service Management:

Services can be managed via the webserver or command line:

```
# Check service status
sudo systemctl status rovo_platform.service

# Start a service
sudo systemctl start rovo_platform.service

# Stop a service
sudo systemctl stop rovo_platform.service

# Enable service at boot
sudo systemctl enable rovo_platform.service

# Disable service at boot
sudo systemctl disable rovo_platform.service
```

Available Services:

Service	Description

rovo_platform.service	Platform driver (CAN, odometry, IMU)
rovo_webserver.service	Web control interface
rovo_controller.service	Joystick and twist mux

Environment Configuration

Setup Script (config/setup.bash):

This script sets up environment variables and aliases.

Environment Variables:

Variable	Default	Description
ROVO_NS	rovo_unit_071	Robot namespace

Useful Aliases:

```
# Build and source workspace
rovo_build
```

CAN Interface Setup

The CAN interface is configured via the rovo_can.bash script:

```
sudo ip link set can0 down
sudo ip link set can0 type can bitrate 500000
sudo ip link set can0 up
```

udev Rules

Hardware udev rules are installed at /etc/udev/rules.d/45-mbs.rules for consistent device naming.

Webserver

Humble

Coming Soon

Jazzy

Robot Webserver

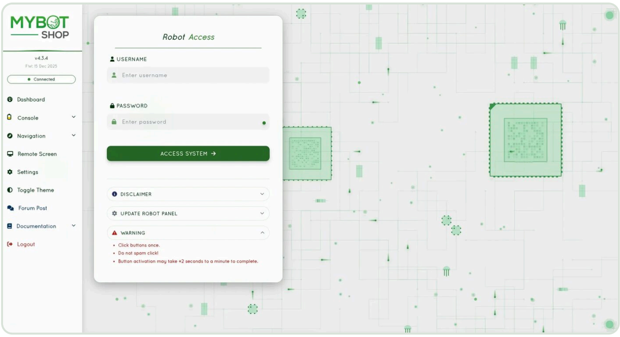
The ROVO3 features a web-based control interface built with Flask.

Access: http://192.168.131.1:9000

Default Credentials:

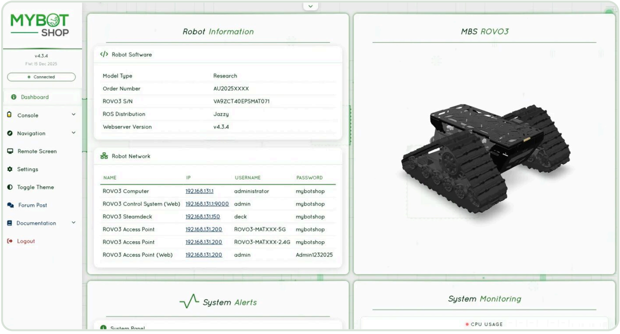
- Username: admin
- Password: mybotshop

Login



Webserver Login Page

Dashboard

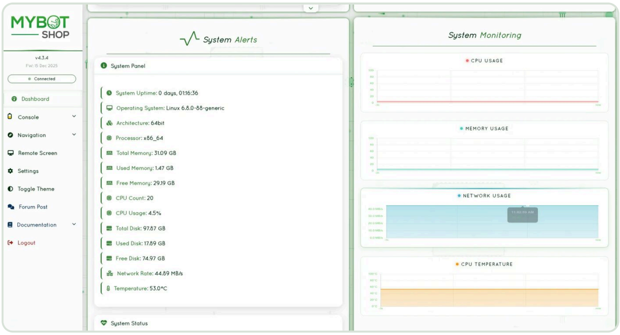


Main Dashboard with 3D Robot Visualization

Features:

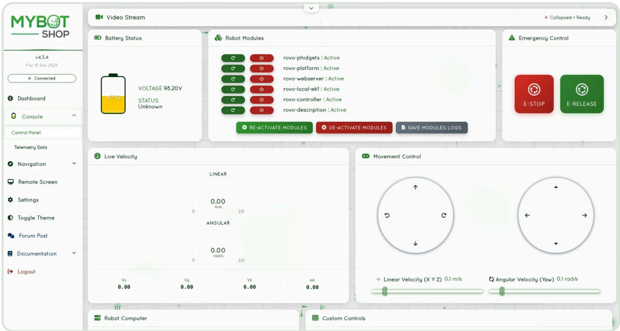
Feature	Description
3D Visualization	Real-time robot model via GLTF
Joystick Teleop	Browser-based joystick control
Service Management	Start/stop ROS2 nodes and services
Map Visualization	View navigation maps
Battery Monitoring	Real-time battery state
GPS Waypoints	Record and manage GPS waypoints
Rosbag Recording	Start/stop rosbag recordings
VNC Access	Remote desktop via VNC

System Management



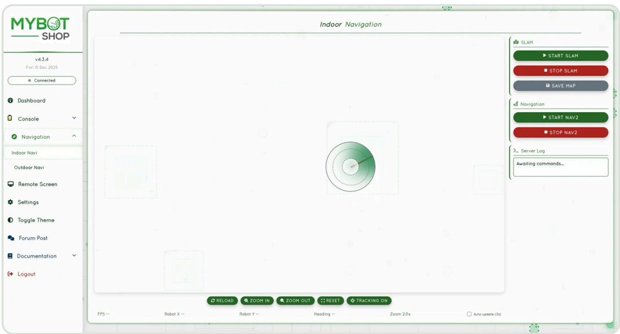
System Management Interface

Console



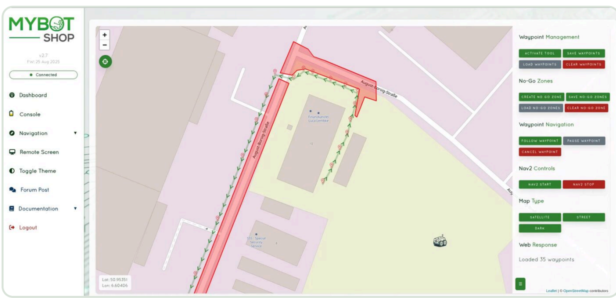
Web Console Interface

Indoor Navigation



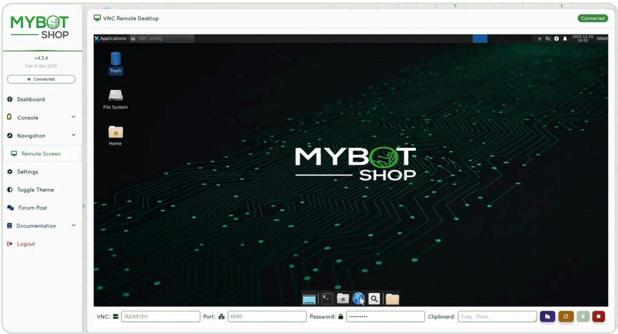
Indoor Navigation View

Outdoor Navigation



Outdoor Navigation View with GPS

VNC Remote Desktop



VNC Remote Desktop Access

Configuration (config/robot_webserver.yaml):

Parameter	Description
robot_rosbag_dir	Rosbag storage directory
robot_services	List of controllable services
robot_map_topic	Map topic name
robot_cmd_vel	Velocity command topic
robot_e_stop	Emergency stop topic
robot_gps_topic	GPS topic
robot_battery_topic	Battery state topic

Launch Webserver:

The webserver is typically started automatically via systemd. To launch manually:

```
ros2 launch rovo_webserver webserver.launch.py
```

Dependencies:

```
pip3 install Flask playsound TTS
sudo apt-get install espeak-ng
```

VNC Setup:

To enable VNC remote desktop:

```
vncpasswd ~/.vnc/passwd
# Password: mybotshop
```

Firmware Updates:

The webserver supports firmware updates for the ROS2 workspace:

1. Navigate to the Firmware Update section
2. Upload a .zip or .tar.gz archive containing the new workspace
3. The system automatically: Creates a backup of the current workspace Extracts and replaces the old workspace Rebuilds using colcon build
4. Restart ROS2 services after update

Teleoperation

Humble

Command Line Interface

Jazzy

Command Line Interface

Teleoperate the ROVO3 using the keyboard via ROS2:

> **Warning:** Extreme caution required when using the ROVO3. Please go through the provided manuals before operating.

Prerequisites:

- Ensure rovo_autostart services are running (check via webserver)
- Robot platform drivers must be active
- Do not run duplicate drivers

1. Set Gear

Before moving the robot, set the appropriate gear:

```
# Set gear to 1 (drive mode)
ros2 service call /$ROVO_NS/platform/set_gear rovo_interface/srv/RovoGear "{gear: 1}"
```

2. Launch Keyboard Teleop

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args --remap
cmd_vel:=/$ROVO_NS/cmd_vel
```

3. Control Keys

Key	Action
i	Move forward
k	Stop
,	Move backward

j	Turn left
l	Turn right
u	Forward + left
o	Forward + right
m	Backward + left
.	Backward + right
q/z	Increase/decrease speed

4. Return to Neutral

When finished, set the gear back to neutral:

```
ros2 service call /$ROVO_NS/platform/set_gear rovo_interface/srv/RovoGear "{gear: 0}"
```

Joystick Control

The ROVO3 supports Logitech joystick control with the following button mapping:

Button	Function
Button 4 (LB)	Enable movement (hold to move)
Button 5 (RB)	Enable turbo mode (hold for faster speed)
Left Stick Y (Axis 1)	Linear X velocity (forward/backward)
Right Stick X (Axis 3)	Angular Z velocity (turning)
D-Pad Up	Increase gear
D-Pad Down	Decrease gear

Speed Modes:

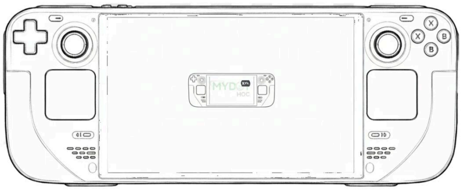
Mode	Linear (m/s)	Angular (rad/s)
Normal (LB)	0.2	0.3
Turbo (LB+RB)	1.0	1.0

Web Interface Control

Access the webserver at http://192.168.131.1:9000 for browser-based joystick control.

Steamdeck Control

The ROVO3 can be controlled via Steamdeck using the steamdeck_joy_teleop interface.



Steamdeck Controller Interface

Visualization

Humble

RViz2

Jazzy

RViz2

RViz2 is used for visualizing the robot state, sensor data, and navigation information.

ROVO3 TF Frame Visualization in RViz2

Launch RViz2:

```
ros2 launch rovo_viz view_robot.launch.py
```

Available Configurations:

Configuration	Description
robot.rviz	Full robot visualization with navigation
model.rviz	Robot model only
steam.rviz	Steamdeck optimized view

Displayed Elements:

- Robot model (URDF)
- TF transforms
- Odometry

- LiDAR point cloud
- Depth camera images
- Navigation costmaps
- Path planning visualization

External Host Setup:

To visualize the robot from an external computer:

1. Build visualization packages: colcon build --symlink-install --packages-select rovo_description rovo_viz source install/setup.bash
2. Set the robot namespace: export ROVO_NS = "rovo_unit_071"
3. Launch RViz2: ros2 launch rovo_viz view_robot.launch.py

> **Note:** Ensure your computer is on the same network as the robot and ROS2 DDS discovery is configured properly.

Rigs

Humble

Coming Soon

Jazzy

MBS Rig System

The ROVO3 platform supports the MBS Rig System for mounting sensors and accessories.

Rig integration documentation coming soon.

> **Note:** Contact MYBOTSHOP for information about available rig configurations for the ROVO3 platform.

Manipulation

Humble

Coming Soon

Jazzy

Robotic Arm Integration

The ROVO3 platform supports integration with robotic manipulators for mobile manipulation applications.

Manipulation documentation coming soon.

Note: Contact MYBOTSHOP for information about supported robotic arm configurations for the ROVO3 platform.

6 ROS 2 Navigation, Sensors & Simulation

Navigation

Humble

Simultaneous Localization and Modulation

Odometry Navigation

Map Navigation

GPS Navigation

Jazzy

Simultaneous Localization and Mapping

SLAM allows the robot to build a map while localizing itself within it.

Prerequisites:

- Ensure rovo_autostart services are running
- LiDAR sensor must be active

Launch SLAM:

```
ros2 launch rovo_navigation slam.launch.py
```

Mapping Procedure:

1. Launch SLAM as shown above
2. Use teleop to drive the robot slowly (recommended: 0.2 m/s)
3. Cover all areas you want to map
4. Ensure good loop closures by revisiting previously mapped areas

Save the Map:

Once satisfied with the map, save it:

```
ros2 run nav2_map_server map_saver_cli -f
/opt/mybotshop/src/mybotshop/rovo_navigation/maps/custom_map \
  --ros-args --remap map:=/$ROVO_NS/map
```

Rebuild Workspace (if using custom map name):

```
cd /opt/mybotshop && colcon build --symlink-install
source /opt/mybotshop/install/setup.bash
```

Odometry Navigation

Odometry navigation allows the robot to navigate using only odometry data, without a pre-built map.

Launch Odometry Navigation:

```
ros2 launch rovo_navigation odom_navi.launch.py
```

This mode is useful for:

- Simple point-to-point navigation
- Testing navigation in unknown environments
- Scenarios where mapping is not required

Map Navigation

Map-based navigation provides full autonomous navigation using a pre-built map.

Prerequisites:

- A saved map must exist
- Map must be built and available in the ROS package

Launch Map Navigation:

```
ros2 launch rovo_navigation map_navi.launch.py
```

Setting Goals:

Goals can be set via:

1. RViz2 “2D Goal Pose” tool
2. Command line
3. Webserver interface

Configuration Files:

- param/nav2_slam.yaml - SLAM parameters
- param/nav2_odom.yaml - Odometry navigation parameters
- param/nav2_map.yaml - Map navigation parameters
- behavior_trees/nav_to_pose.xml - Custom behavior tree

Dependencies:

```
sudo apt-get install ros-jazzy-navigation2 ros-jazzy-nav2-bringup ros-jazzy-twist-mux
```

For more information, see the Nav2 documentation .

GPS Navigation

GPS-based navigation is available for outdoor operations. Configure based on your GPS hardware and use the webserver for waypoint recording and playback.

Sensors

Humble

Depth Cameras

Lidars

GPS

Auxiliary

Jazzy

Depth Cameras

The ROVO3 supports the Intel RealSense D435i depth camera for RGB-D perception.

Test Native Driver:

```
ros2 launch realsense2_camera rs_launch.py depth_module.depth_profile:=1280x720x30
pointcloud.enable:=true
```

Launch via ROVO3 Package:

The camera can be controlled via the webserver or systemctl. When disabled via services, test manually:

```
ros2 launch rovo_depth_camera realsense.launch.py
```

Published Topics:

Topic	Type
/camera/color/image_raw	sensor_msgs/Image
/camera/depth/image_rect_raw	sensor_msgs/Image
/camera/depth/color/points	sensor_msgs/PointCloud2

> **Note:** The ROVO3 utilizes the camera's inbuilt IMU. External USB connection may be required for depth functionality.

Lidars

The ROVO3 supports Ouster OS1-64 LiDAR for 3D perception and mapping.

Network Configuration:

- Default IP: 192.168.131.20

Launch via ROVO3 Package:

The LiDAR can be controlled via the webserver or systemctl. When disabled via services, test manually:

```
ros2 launch rovo_lidar ouster.launch.py
```

Published Topics:

Topic	Type
/ouster/points	sensor_msgs/PointCloud2
/ouster/scan	sensor_msgs/LaserScan
/ouster/imu	sensor_msgs/Imu

Dependencies:

```
# Install Ouster ROS2 driver
sudo apt-get install ros-jazzy-ouster-ros
```

For more information, see the Ouster ROS2 driver documentation .

GPS

GPS integration is available for outdoor navigation and waypoint recording. Configuration depends on the specific GPS hardware installed on your ROVO3 unit.

Auxiliary

The ROVO3 platform includes an integrated IMU that publishes orientation and acceleration data.

Published Topic:

- /<ns>/platform/imu/data (sensor_msgs/Imu)

The BMS provides battery state information.

Published Topic:

- /<ns>/platform/bms/state (sensor_msgs/BatteryState)

Simulation

Humble

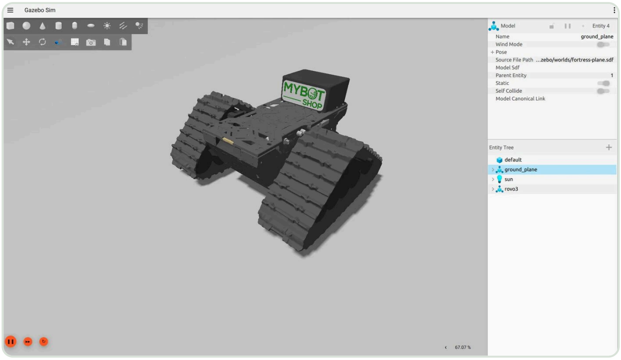
Gazebo

Isaac Sim

Jazzy

Gazebo

The ROVO3 uses Gazebo Harmonic for simulation, which is the recommended simulator for ROS2 Jazzy.



ROVO3 Simulation in Gazebo Harmonic

Launch Simulation:

```
ros2 launch rovo_gazebo simulation.launch.py
```

Available Worlds:

World File	Description
fortress-plane.sdf	Flat ground plane (default)
fortress-substation.sdf	Industrial substation environment
fortress-moon.sdf	Lunar terrain

Teleoperate in Simulation:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Simulated Sensor Topics:

Topic	Type	Description
/clock	rosgraph_msgs/Clock	Simulation time
/odom	nav_msgs/Odometry	Robot odometry
/sensors/imu/data	sensor_msgs/Imu	IMU data
/sensors/d435i/image	sensor_msgs/Image	RGB camera
/sensors/d435i/depth_image	sensor_msgs/Image	Depth image
/sensors/d435i/points	sensor_msgs/PointCloud2	RGB-D point cloud

/sensors/os_sensor/scan	sensor_msgs/LaserScan	LiDAR scan
/sensors/os_sensor/points	sensor_msgs/PointCloud2	LiDAR point cloud

Clean Up Gazebo Processes:

If Gazebo does not close properly, use the cleanup script:

```
ros2 run rovo_gazebo kill_gz.sh
```

Dependencies:

```
sudo apt-get install ros-jazzy-ros-gz ros-jazzy-ros-gz-bridge
```

For more information, see the Gazebo Harmonic documentation .

Isaac Sim

NVIDIA Isaac Sim integration is planned for future releases. Check back for updates on Isaac Sim support for the ROVO3 platform.

7 ROS 2 Package Reference, Debugging & Installation

Packages

Humble

Robot Description

Twist Mux

EKF Localization

Jazzy

rovo_description

URDF and Xacro robot description files for the ROVO3 tracked robot platform.

Contents:

- launch/rovo_description.launch.py - Robot state publisher
- launch/view_robot.launch.py - View URDF in RViz
- meshes/ - Platform and sensor visual meshes
- xacro/robot.xacro - Main robot description

Usage:

```
# Publish robot description
ros2 launch rovo_description rovo_description.launch.py

# Get URDF from Xacro
xacro $(ros2 pkg prefix rovo_description)/share/rovo_description/xacro/robot.xacro
```

rovo_platform

Hardware driver and control interface for the ROVO3 tracked robot platform. Provides CAN bus communication, motor control, odometry publishing, and ROS2 service interfaces.

Configuration Parameters (config/rovo_platform.yaml):

Parameter	Default	Description
node_id	2	CAN node ID
can_channel	can0	CAN interface name
bitrate	500000	CAN bitrate (500 kbit/s)
wheel_separation	0.92	Distance between tracks (m)
max_speed_mps	1.0	Maximum linear speed (m/s)

drive_diameter	0.35	Drive sprocket diameter (m)
gear_ratio	16.0	Gear ratio (16.0 for i16, 7.0 for i7)
publish_tf	true	Publish TF transforms

Published Topics:

Topic	Type	Description
/<ns>/platform/odom	nav_msgs/Odometry	Robot odometry
/<ns>/platform/joint_states	sensor_msgs/JointState	Track joint states
/<ns>/platform/imu/data	sensor_msgs/Imu	IMU data
/<ns>/platform/bms/state	sensor_msgs/BatteryState	Battery state

Subscribed Topics:

Topic	Type	Description
/<ns>/platform/cmd_vel	geometry_msgs/Twist	Velocity commands

Services:

Service	Type	Description
/<ns>/platform/set_gear	rovo_interface/RovoGear	Set gear (0=neutral, 1-3=drive)
/<ns>/platform/set_mode	rovo_interface/RovoModes	Set operating mode

Usage:

```
ros2 launch rovo_platform rovo_driver.launch.py

# Set gear
ros2 service call /$ROVO_NS/platform/set_gear rovo_interface/srv/RovoGear "{gear: 1}"
```

rovo_controller

Joystick teleoperation and command multiplexing for the ROVO3 platform.

Twist Mux Priority Table:

Priority	Source	Topic
255	E-Stop (Lock)	e_stop
25	Logitech Joystick	joy_teleop/cmd_vel
20	Steamdeck	steamdeck_joy_teleop/cmd_vel

15	Web Interface	webserver/cmd_vel
10	Interactive Marker	twist_marker_server/cmd_vel
5	External	cmd_vel
3	Autonomous (High)	autonomous_high_priority/cmd_vel
2	Autonomous (Mid)	autonomous_mid_priority/cmd_vel
1	Autonomous (Low)	autonomous_low_priority/cmd_vel

Logitech Joystick Button Mapping:

Button	Function
Button 4 (LB)	Enable movement (hold to move)
Button 5 (RB)	Enable turbo mode (hold for faster speed)
Left Stick Y (Axis 1)	Linear X velocity (forward/backward)
Right Stick X (Axis 3)	Angular Z velocity (turning)
D-Pad Up	Increase gear
D-Pad Down	Decrease gear

Speed Settings:

Mode	Linear (m/s)	Angular (rad/s)
Normal	0.2	0.3
Turbo	1.0	1.0

rovo_gazebo

Gazebo Harmonic simulation package for the ROVO3 platform.

Available Worlds:

- fortress-plane.sdf - Flat ground plane (default)
- fortress-substation.sdf - Industrial substation environment
- fortress-moon.sdf - Lunar terrain

Simulated Topics (GZ -> ROS):

Topic	Type	Description
/clock	rosgraph_msgs/Clock	Simulation time
/odom	nav_msgs/Odometry	Robot odometry
/sensors/imu/data	sensor_msgs/Imu	IMU data

/sensors/d435i/image	sensor_msgs/Image	RGB camera
/sensors/d435i/depth_image	sensor_msgs/Image	Depth image
/sensors/d435i/points	sensor_msgs/PointCloud2	RGB-D point cloud
/sensors/os_sensor/scan	sensor_msgs/LaserScan	LiDAR scan
/sensors/os_sensor/points	sensor_msgs/PointCloud2	LiDAR point cloud

rovo_navigation

Nav2-based navigation stack for the ROVO3 platform.

Launch Files:

Launch File	Description
slam.launch.py	SLAM mapping mode
odom_navi.launch.py	Odometric navigation (no map)
map_navi.launch.py	Map-based autonomous navigation

Configuration Files:

- param/nav2_slam.yaml - SLAM parameters
- param/nav2_odom.yaml - Odometry navigation parameters
- param/nav2_map.yaml - Map navigation parameters
- behavior_trees/nav_to_pose.xml - Custom behavior tree

rovo_viz

RViz2 visualization package with pre-configured displays.

RViz Configurations:

- rviz/robot.rviz - Full robot visualization with navigation
- rviz/model.rviz - Robot model only
- rviz/steam.rviz - Steamdeck optimized view

rovo_autostart

System bringup and automatic startup configuration.

Files:

- config/setup.bash - Environment variables and aliases
- config/startup.bash - Startup script
- config/rovo_can.bash - CAN interface setup
- debian/45-mbs.rules - udev rules for hardware
- scripts/startup_installer.py - Systemd service installer

Environment Variables:

Variable	Default	Description
ROVO_NS	rovo_unit_071	Robot namespace

rovo_webserver

Web-based robot control interface with Flask backend.

Features:

- Real-time robot visualization (3D model via GLTF)
- Joystick teleoperation
- Service management (start/stop ROS nodes)
- Map visualization
- Battery monitoring
- GPS waypoint recording
- Rosbag recording
- VNC remote desktop access

Access: http://192.168.131.1:9000

rovo_interface

Custom ROS2 service definitions for the ROVO3 platform.

RovoGear.srv:

```
int32 gear      # 0=neutral, 1-3=drive gears
---
bool success
string message
```

RovoModes.srv:

```
string request_data
---
bool success
string reason
```

Usage:

```
# Set gear to 1
ros2 service call /$ROVO_NS/platform/set_gear rovo_interface/srv/RovoGear "{gear: 1}"

# Set mode
ros2 service call /$ROVO_NS/platform/set_mode rovo_interface/srv/RovoModes "{request_data:
'auto'}"
```

Debugging

Humble

Coming Soon

Jazzy

RQT

RQT provides a GUI framework for ROS2 debugging tools.

Launch RQT:

```
ros2 run rqt_gui rqt_gui --ros-args --remap tf:=/$ROVO_NS/tf --remap
tf_static:=/$ROVO_NS/tf_static
```

Available Plugins:

- Topic Monitor
- Service Caller
- Parameter Reconfigure
- Node Graph
- TF Tree

Dynamic Reconfigure

Use dynamic reconfigure to adjust parameters at runtime:

```
ros2 run rqt_gui rqt_gui --ros-args --remap tf:=/$ROVO_NS/tf --remap
tf_static:=/$ROVO_NS/tf_static
```

From the GUI, select Plugins > Configuration > Dynamic Reconfigure .

TF

TF Tree Visualization:

```
ros2 run rqt_tf_tree rqt_tf_tree --force-discover \
  --ros-args --remap tf:=/$ROVO_NS/tf --remap tf_static:=/$ROVO_NS/tf_static
```

Generate TF Frames PDF:

```
ros2 run tf2_tools view_frames.py --force-discover \
  --ros-args --remap tf:=/$ROVO_NS/tf --remap tf_static:=/$ROVO_NS/tf_static
```

Topic Debugging

List all topics:

```
ros2 topic list
```

Echo a topic:

```
ros2 topic echo /$ROVO_NS/platform/odom
```

Check topic frequency:

```
ros2 topic hz /$ROVO_NS/platform/odom
```

Service Debugging

List all services:

```
ros2 service list
```

Call a service:

```
ros2 service call /$ROVO_NS/platform/set_gear rovo_interface/srv/RovoGear "{gear: 0}"
```

Node Debugging

List all nodes:

```
ros2 node list
```

Get node info:

```
ros2 node info /rovo_platform
```

CAN Debugging

Configure CAN Interface:

```
sudo ip link set can0 down
sudo ip link set can0 type can bitrate 500000
sudo ip link set can0 up
```

Monitor CAN Traffic:

```
sudo candump can0
```

Note: The CAN bitrate for ROVO3 is 500000 (500 kbit/s).

Common Troubleshooting

Service Not Found

If you see “Unit ros2.service could not be found”, it means there is no startup service installed. Run:

```
ros2 run rovo_autostart startup_installer.py
```

Service Status Indicators

- Red marker - The startup job has failed
- Green marker - Everything is working correctly
- Grey marker - The service has not started yet

Restart Failed Service

```
sudo service rovo-platform restart
```

Network Connection Issues

> **Note:** Sometimes other networks can cause disruptions when connecting to the ROVO3. It is best to have only your connection to the robot active and all others inactive.

Miscellanious

Humble

Coming Soon

Jazzy

Sync Host and Robot

Synchronize files between host computer and robot using rsync:

Sync to Robot:

```
rsync -avP -t --delete -e ssh src robot@192.168.131.1://opt/mybotshop
```

Sync to Steamdeck:

```
rsync -avP -t --delete -e ssh src deck@192.168.131.150://home/deck/ros2_ws
```

Core Launch Files

> **Warning:** Do not run these files manually as they should already be running via the webserver!

These launch files are managed by systemd services and the webserver:

Platform Driver:

```
ros2 launch rovo_platform rovo_driver.launch.py
```

Webserver:

```
ros2 launch rovo_webserver webserver.launch.py
```

Controller:

```
ros2 launch rovo_controller controller.launch.py
```

Robot Description:

```
ros2 launch rovo_description rovo_description.launch.py
```

CAN Hardware Connections

Recommended Parts:

- Primary: Part 1-1564337-1
- Backup: Part 967650-1

Pin Mapping:

ROVO	Peak-USB
Pin1 Can2 Low	Pin2 Can Low
Pin2 Can2 High	Pin7 Can High
Pin3 Can Gnd	Pin3 Can Gnd

Useful Commands

Build Workspace:

```
cd /opt/mybotshop
colcon build --symlink-install --cmake-args -DCMAKE_BUILD_TYPE=Release
source install/setup.bash
```

Quick Build Alias:

```
rovo_build
```

Check ROS2 Environment:

```
printenv | grep ROS
```

Source Workspace:

```
source /opt/mybotshop/install/setup.bash
```

Installation

Humble

Installation Robot

Installation External

Jazzy

Installation Robot

The ROVO3 comes pre-installed with ROS2 Jazzy. For a fresh installation or re-installation, follow these steps:

1. Set Robot Hostname

```
sudo hostnamectl set-hostname rovo-unit-071
```

2. Create Workspace Directory

```
sudo mkdir /opt/mybotshop && sudo chown -R robot:robot /opt/mybotshop
```

3. Clone and Install

Clone the repository to the robot's PC and run the installer:

```
cd /opt/mybotshop/src/mybotshop/robot_installer
sudo chmod +x total_install.sh
sudo ./total_install.sh
```

4. Build ROS2 Workspace

```
cd /opt/mybotshop
colcon build --symlink-install --cmake-args -DCMAKE_BUILD_TYPE=Release
source install/setup.bash
```

5. Install Startup Services

```
ros2 run rovo_autostart startup_installer.py
```

6. Disable System Suspend

```
sudo systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
sudo systemctl disable systemd-networkd-wait-online.service
sudo systemctl mask systemd-networkd-wait-online.service
```

7. Configure Network

Edit the netplan configuration:

```
sudo nano /etc/netplan/01-network-manager.yaml
```

Add the following configuration:

```
network:
  version: 2
  renderer: NetworkManager
  bridges:
    br0:
      interfaces:
        - eno1
        - enp3s0
      dhcp4: no
      addresses:
        - 192.168.131.1/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
      parameters:
        stp: false
        forward-delay: 0
      optional: true

  ethernets:
    eno1:
      dhcp4: no
      optional: true
    enp1s0:
      dhcp4: no
      optional: true
```

Apply the configuration:

```
sudo netplan apply
```

Installation External

For external (host) computers that need to visualize and interact with the ROVO3:

1. Clone the Repository

Clone the ROVO3 repository into your ROS2 workspace.

2. Build Essential Packages

```
colcon build --symlink-install --packages-select rovo_description rovo_viz
source install/setup.bash
```

3. Set Environment and Launch

```
export ROVO_NS="rovo_unit_071"
ros2 launch rovo_viz view_robot.launch.py
```

Dependencies

Install the required ROS2 packages:

```
sudo apt-get install ros-jazzy-urdf ros-jazzy-xacro \
                     ros-jazzy-joint-state-publisher ros-jazzy-robot-state-publisher \
                     ros-jazzy-joy ros-jazzy-teleop-twist-joy ros-jazzy-twist-mux \
                     ros-jazzy-navigation2 ros-jazzy-nav2-bringup \
                     ros-jazzy-ros-gz ros-jazzy-ros-gz-bridge
```

8 Safety Guidelines & Maintenance

Note: The Robotic Manipulator Safety section below is reproduced verbatim from the ROVO3 source documentation. ROVO3 ships as a bare tracked base with no manipulator fitted — this section appears to be boilerplate shared across MYBOTSHOP/QUADRUPED robot manuals and is retained here for completeness rather than silently removed. It becomes directly relevant if a robotic arm is added for mobile manipulation, as mentioned in Chapter 3.

Robotic Manipulator Safety

When deploying robotic manipulators, it is imperative to prioritize safety procedures to mitigate risks and ensure secure operations. The following guidelines delineate key safety measures when working with robotic manipulators.

Work Area Safety

- Maintain a clean and well-organized work area. Cluttered or inadequately lit spaces can impede the proper functioning of sensors and hinder precise manipulation.
- Avoid operating robotic manipulators in hazardous environments, such as those containing corrosive substances, extreme temperatures, or sharp objects that may damage the manipulator.
- Ensure that only authorized personnel are present in the vicinity during manipulator operation to prevent interference and ensure a safe working environment.

Electrical Safety

- Ensure the manipulator's power system adheres to electrical safety standards. Regularly inspect and maintain power components to prevent malfunctions.
- Implement safeguards to protect the manipulator from adverse environmental conditions, such as exposure to moisture or extreme temperatures.
- Regularly inspect power cables and connections, promptly replacing damaged components to minimize the risk of electrical issues.

Manipulation Safety

- Implement collision detection systems to prevent unintended contact with objects, humans, or other equipment during manipulation tasks.
- Define and enforce safety zones around the manipulator's workspace to minimize the risk of unintended interactions with personnel or other objects.
- Regularly calibrate and test the manipulator's sensors and systems to ensure precise and reliable performance during manipulation tasks.

Emergency Response

- Install an emergency stop mechanism to swiftly halt manipulator operation in unforeseen circumstances or emergencies.
- Clearly mark and communicate emergency stop locations within the manipulator's operational area.
- Conduct regular emergency response drills to ensure personnel are familiar with procedures for handling unexpected situations during manipulator operation.

Data Security and Privacy

- Implement robust cybersecurity measures to safeguard the manipulator's control systems and data from unauthorized access or manipulation.
- Ensure compliance with privacy regulations when collecting, storing, or transmitting data captured by the manipulator's sensors.

Human Interaction Safety

- Integrate sensors and communication systems to detect and respond to the presence of humans in the manipulator's vicinity.
- Clearly communicate the manipulator's operational status and intentions using visual and audible signals to alert nearby individuals.
- Establish protocols for safe human-robot collaboration, particularly in shared workspaces where manipulators are in operation.

Residual Risks

Despite the implementation of safety measures, certain residual risks may persist. These include:

- Impairment of sensor functionality.
- Risk of collisions during complex manipulation tasks.
- Cybersecurity vulnerabilities.
- Unintended human interactions due to unforeseen circumstances.

Robotic manipulators are sophisticated technologies that demand correct usage to avoid accidents and ensure a secure environment. Please adhere to proper procedures diligently, prioritizing both precision and safety.

Autonomous Robot Safety

When deploying autonomous mobile robots, prioritizing safety procedures is essential to prevent accidents and ensure secure operations. The following guidelines outline key safety measures when working with an autonomous mobile robot:

Work Area Safety

Maintain a clean and well-lit work area. Cluttered or poorly illuminated spaces can impede the proper functioning of sensors and navigation systems.

- Avoid operating autonomous mobile robots in explosive atmospheres, such as areas with flammable liquids, gases, or dust. The robot's components may pose a risk in such environments.
- Keep bystanders and unauthorized personnel at a safe distance during robot operation to prevent interference with autonomous navigation.

Electrical Safety

- Ensure the robot's power system adheres to electrical safety standards. Regularly inspect and maintain power components to prevent malfunctions.
- Implement mechanisms to protect the robot from adverse weather conditions, such as rain or wet environments.
- Regularly inspect power cables and connections and replace damaged components promptly to minimize the risk of electrical issues.

Navigation Safety

- Implement obstacle detection and avoidance systems to prevent collisions with objects, people, or other robots.
- Define and enforce safety zones within the robot's operational area to minimize the risk of unintended interactions with personnel or other equipment.
- Regularly calibrate and test the robot's navigation sensors and systems to ensure accurate and reliable performance.

Emergency Response

- Install an emergency stop mechanism to quickly halt the robot's operation in case of unforeseen circumstances or emergencies.
- Clearly mark and communicate emergency stop locations throughout the robot's operational area.
- Conduct regular emergency response drills to ensure personnel are familiar with procedures for handling unexpected situations.

Data Security and Privacy

- Implement robust cybersecurity measures to protect the robot's control systems and data from unauthorized access or manipulation.
- Ensure compliance with privacy regulations when collecting, storing, or transmitting data captured by the robot's sensors.

Human Interaction Safety

- Integrate sensors and communication systems to detect and respond to the presence of humans in the robot's vicinity.
- Clearly communicate the robot's operational status and intentions using visual and audible signals to alert nearby individuals.
- Establish protocols for safe human-robot collaboration, especially in shared workspaces.

Residual Risks

Despite the implementation of safety measures, certain residual risks may persist. These include:

- Impairment of sensor functionality.
- Risk of collisions in crowded or dynamic environments.
- Cybersecurity vulnerabilities.
- Unintended human interactions due to unforeseen circumstances.

Autonomous Mobile Robots (AMR) are advanced technologies that require correct usage to avoid accidents and ensure a secure environment. Learn and follow the proper procedures diligently; prioritize both quality and safety.

Robot Maintenance

Regular maintenance is essential to ensure the safe and reliable operation of the ROVO3 tracked robot platform.

Pre-Operation Checks

Before each operation, perform the following checks:

- Verify that the emergency stop buttons are functioning correctly on both the robot and remote controller
- Inspect the tracks for damage, wear, or debris
- Check that all mounting hardware is secure
- Verify battery charge level is sufficient for planned operation
- Ensure all sensors are clean and unobstructed
- Test the communication link between the robot and remote controller

Track System Maintenance

- Regularly inspect track tension and adjust as needed according to manufacturer specifications
- Check for wear on track pads and sprockets
- Remove debris, mud, or foreign objects from the track system after each use
- Inspect drive sprockets for signs of wear or damage
- Lubricate moving parts as specified in the manufacturer documentation

Electrical System Maintenance

- Inspect all cable connections for wear, damage, or corrosion
- Check power connectors and terminals for secure connections
- Verify CAN bus connections are properly seated
- Inspect the battery for physical damage or swelling
- Clean electrical contacts periodically to prevent oxidation

Battery Maintenance

- Follow the manufacturer's charging guidelines
- Do not overcharge or deep discharge the battery
- Store the battery at recommended charge levels when not in use
- Keep battery terminals clean and free of corrosion
- Monitor battery health through the BMS (Battery Management System)
- Replace batteries that show signs of degradation or reduced capacity

Sensor Maintenance

- Keep LiDAR lenses clean and free of dust, dirt, or fingerprints
- Clean camera lenses with appropriate lens cleaning materials
- Verify sensor mounting positions remain calibrated
- Check sensor cables for wear or damage
- Test sensor functionality periodically

Software Maintenance

- Keep the ROS2 software stack updated with the latest patches
- Regularly backup configuration files and maps
- Monitor system logs for errors or warnings
- Verify autostart services are functioning correctly
- Test emergency stop functionality after software updates

Storage and Transport

- Store the robot in a dry, temperature-controlled environment
- Protect sensors from dust and physical damage during storage
- Charge the battery to recommended storage levels (typically 50-60%)
- Secure the robot during transport to prevent damage
- Use original packaging or appropriate protective cases when shipping

Maintenance Schedule

Task	Frequency	Notes
Pre-operation checks	Before each use	Required
Track inspection	Weekly	More frequent in harsh conditions
Sensor cleaning	Weekly	As needed based on environment
Cable inspection	Monthly	Check for wear and damage
Battery health check	Monthly	Monitor through BMS
Firmware updates	As released	Test in safe environment first
Full system inspection	Quarterly	Comprehensive check of all systems

Warning: Always power off the robot and disconnect the battery before performing any maintenance on electrical or mechanical systems. Only qualified personnel should perform repairs or modifications to the robot.

9 Getting Help & RMA

Note: The source documentation for this section carries a couple of leftover-boilerplate inconsistencies, reproduced faithfully below rather than silently corrected: the RMA email step references `support@quadruped.de` and `docs.quadruped.de` (a sister MYBOTSHOP brand's addresses), and the Resources card titled "Forum" actually contains Newsletter sign-up content and a Newsletter QR code — a correctly-labeled Forum card appears further down, under Contact Information. **For ROVO3, use the MYBOTSHOP-branded contacts throughout this manual and on the docs page: support@mybotshop.de, forum.mybotshop.de.**

RMA / Support

How to Raise a Return Merchandise Authorization (RMA) Case

At QUADRUPED Robotics, we are committed to providing excellent support to ensure your issues are resolved promptly. If you encounter any problems with our products, please follow the steps below to raise a Return Merchandise Authorization (RMA) case:

Step 1: Open a Topic in Our Forum

1. Visit MYBOTSHOP Forum .
2. Open a new topic describing your issue in detail. Our support team and community members will attempt to assist you in resolving the problem remotely.

Step 2: Contact Support for RMA Number

If the issue cannot be resolved remotely through our forum:

1. Email us at support @ quadruped . de .
2. Provide a detailed description of the issue and include any troubleshooting steps already taken.
3. Our support team will evaluate your case and, if necessary, issue a RMA number along with a return address or a shipping label.

Important Shipping Information

Note: Please note: All goods must be sent back to the following address: QUADRUPED Robotics GmbH c / o MYBOTSHOP GmbH Willy - Messerschmitt - Strasse 12 50126 Bergheim Germany Do not ship goods without obtaining a RMA number. Parcels sent without a RMA number will be declined and returned to the sender. Do not ship goods to QUADRUPED Robotics GmbH Office in Leverkusen!

Resources

For your convenience, we have compiled the most important resources related to the robot platform. Each link can be accessed directly or by scanning the corresponding QR code.

Resources

For your convenience, we have compiled the most important resources related to the robot platform. Each link can be accessed directly or by scanning the corresponding QR code.

Documentation

Our official documentation contains detailed setup instructions, usage guidelines, and troubleshooting information. Access the latest version here: https://www.docs.quadruped.de/index.html

Forum

Stay informed about product updates, events, and special offers by subscribing here: https://www.mybotshop.de/Newsletter_1

Contact Information

Forum

Connect with other developers and researchers in our community forum. Here you can ask questions, share experiences, and stay up to date: https://forum.mybotshop.de/

Rentals

If you would like to test or temporarily use a robot, we also offer rental services. Browse our available systems here: https://www.mybotshop.de/Vermietung_3

Contact Information

- Sales 📞 +49 (0) 2271 588 93 10 | ✉️ sales @ mybotshop . de
- After Sales 📞 +49 (0) 2271 588 93 20 | ✉️ support @ mybotshop . de
- Accounting 📞 +49 (0) 2271 588 93 80 | ✉️ accounting @ mybotshop . de
- General inquiries 📞 +49 (0) 2271 588 93 0 | ✉️ info @ mybotshop . de

Frequently Asked Questions

...